Trexquant Hangman Challenge

Subject: Code explain and methods utilization

Submitted By : Rakshith Kumar KN

This part of a word-guessing game or a similar application. Let's break down the approach:

1. **Building n-word dictionary:**
   - The `build_n_word_dictionary` method creates a dictionary (`n_word_dict`) where keys represent the length of words (from 3 to 29), and values are lists of substrings of that length from the given `word_list`.
   - It iterates through each word in `word_list` and generates all possible substrings of lengths 3 to 29.

2. **Finding maximum word length:**
   - The `find_max_length` method determines the maximum length of words in the provided `word_list`.

3. **Building a dictionary from a file:**
   - The `build_dictionary` method reads a dictionary file and returns its contents as a list.

4. **Counting vowels in a word:**
   - The `vowel_count` method calculates the ratio of vowels to the total number of characters in a given word.

5. **Frequency counting functions:**
   - The `func` and `func2` methods seem to be aimed at counting the frequency of letters in a list of words (`new_dictionary`).
   - `func` uses the Counter class to count letter frequencies across all words.
   - `func2` narrows down the dictionary to words matching a pattern (`clean_word`) and then applies `func` to count letter frequencies.

6. **Guessing a letter:**
   - The `guess` method takes a word and the n-word dictionary as input to guess a letter.
   - It first filters the current dictionary based on the length and pattern of the guessed word.
   - It then prioritizes letter guesses based on frequency and certain conditions (e.g., vowel count) and updates the guessed letters accordingly.

7. **Fallback strategies:**
   - If no suitable letter is found through the initial guesses, the code employs additional strategies.
   - It considers substrings of the guessed word and tries to find common letters in those substrings.

   - Finally, if all else fails, it resorts to a default ordering of letters.

8. **Updating guessed letters:**
   - The guessed letter is added to the list of already guessed letters (`self.guessed_letters`).
   - The method returns the guessed letter.

This is a part of a larger application for a word-guessing game, possibly a hangman-type game. The approach involves progressively narrowing down the potential letters based on the length and pattern of the word and the frequency of letters in the dictionary.